Fluid - ViewHelper - Incubator - Suggestion #46218

ImageViewHelper with comma separated images and imageLinkWrap

2013-03-12 20:03 - Andrea Schmuttermair

Status:	New	**Start date:**	2013-03-12
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
Has patch:	No	**Tags:**	

Description

Hi,

This ImageViewHelper can process image lists as they come from database: image1.jpg,image2.jpg

It just calls the standard ImageViewHelper for every image.

Then you can add a ImageLinkWrap config array, so you can use JSWindow.

And every image can be wrapped.

Example:

```
In Controller: 
$imageLinkWrap = array(
        'JSwindow' => 1,
        'JSwindow.' => array(
            'newWindow' => '0',
            'expand' => '17,20',
        ),
        'enable' => 1,
);
$this->view->assign('images', $item->getImages());
$this->view->assign('imageLinkWrap', $imageLinkWrap);

In Fluid Template: 
<myviewhelper:imageLinkWrap srcList="{images}" srcPath="uploads/tx_myextension/" alt="" width="120m" height="120m" imageLinkWrap="{imageLinkWrap}" wrap="<li>|</li>" />
```

View Helper Code:

```
class Tx_MyExtension_ViewHelpers_ImageLinkWrapViewHelper extends Tx_Fluid_ViewHelpers_ImageViewHelper {

    /**
     * Extends ImageViewHelper with: 
     * - image lists (comma-separated image filenames)
     * - image wrap for every list item
     * - imageLinkWrap TypoScript
     *
     * @param string $srcList image list like: image1.jpg,image2.jpg (of course only one image is possible, too!)
     * @param string $srcPath image upload path
     * @param string $width width of the image. This can be a numeric value representing the fixed width of the image in pixels. But you can also perform simple calculations by adding "m" or "c" to the value. See imgResource.width for possible options.
     * @param string $height height of the image. This can be a numeric value representing the fixed height of the image in pixels. But you can also perform simple calculations by adding "m" or "c" to the value. See imgResource.width for possible options.
     * @param integer $minWidth minimum width of the image
     * @param integer $minHeight minimum height of the image
     * @param integer $maxWidth maximum width of the image
     * @param integer $maxHeight maximum height of the image
     * @param array $imageLinkWrap TypoScript imageLinkWrap TypoScript Configuration array
     * @param string $wrap a wrap for every image in the list
     *
     * @return string rendered tag.
     * @author Andrea Schmuttermair <spam@schmutt.de>
     */
    public function render($srcList, $srcPath, $width = NULL, $height = NULL, $minWidth = NULL, $m
inHeight = NULL, $maxWidth = NULL, $maxHeight = NULL, $imageLinkWrap = NULL, $wrap = "") {

        $imageArray = explode(",", $srcList);

        $output = "";
        foreach($imageArray as $image) {

            //create image with fluid image view helper
            $imagetag = parent::render($srcPath . $image, $width, $height, $minWidth, $minHeight,
$maxWidth, $maxHeight );

            //process imageLinkWrap
            $imageOutput = $imagetag;
            if ($imageLinkWrap != NULL) {
                $imageOutput = $this->contentObject->imageLinkWrap($imagetag, $srcPath . $image, $
imageLinkWrap);
            }

            //process wrap
            if (strlen($wrap)>0) {
                $imageOutput = $this->contentObject->wrap($imageOutput,$wrap);
            }
            $output .= $imageOutput;

        }

        return $output;
    }

}
```